METHOD FOR GENERATING A DISPLAY UTILIZING OBJECTS IN AN OBJECT LIST

This application is a continuation of prior application Ser. No. 07/322,740, filed Mar. 13, 1989, now abandoned.

RELATED INVENTIONS

The present invention is related to the following inventions, all assigned to the assignee of the present invention:

System Control Structure of a Hospital Information System and Method of Using Same, invented by John Brimm et al., having U.S. Ser. No. 116,614, and filed on Nov. 3, 1987, U.S. Pat. No. 4,878,175;

Method for Generating Patient-Specific Flowsheets by Adding/Deleting Parameters, invented by Ronald Norden-Paul et al., having U.S. Ser. No. 116,611, and filed on Nov. 3, 1987;

Medical Information System With Automatic Updating of Task List In Response to Charting Interventions On Task List Window Into An Associated Form invented by John Brimm et al., having U.S. Ser. No. 268,822, and filed on Nov. 7, 1988, now U.S. Pat. No. 5,077,666;

Clinical Task List with Charting onto Underlying Form and Automatic Updating of Task List, invented by John Brimm et al., having U.S. Ser. No. 268,323, and filed Nov. 7, 1988, now U.S. Pat. No. 5,072,383;

A Method for Displaying Information from an Information Based Computer System, having Ser. No. 07/407,979 and filed on Sep. 15, 1989;

Spreadsheet Cell having Multiple Data Fields, having Ser. No. 07/408,166 and filed on Sep. 15, 1989;

A Method for Displaying Information from an Information Based Computer System, having Ser. No. 07/407,836 and filed on Sep. 15, 1989;

Electronic Data Storage Interface, having Ser. No. 07/408,178 and filed on Sep. 15, 1989;

Method for Updating Data in a Database, having Ser. No. 07/408,167 and filed on Sep. 27, 1989;

Method for Storing a Transaction in a Distributed Database System, having Ser. No. 07/408,169 and filed on Sep. 15, 1989;

A Method of Forming a Spreadsheet Display, having Ser. No. 07/407,972 and filed on Sep. 15, 1989; and

Data Storage Audit Trail, having Ser. No. 07/409,230 and filed on Sep. 15, 1989.

BACKGROUND OF THE INVENTION

The present invention relates, in general, to a process of controlling the input and output of data in a computing system and, more particularly, to a process of managing the form of input and output with the use of tiles consisting of one or more cells.

In the art, there are various types of input and output buffers. However, this is typically handled using two different buffers, one for inputs and one for outputs. A typical input is generated by some type of user interface such as a keyboard, mouse, track ball, or the like. The output is typically provided on a cathode ray tube (CRT), flat screen display, or like device generally classified as operator display modules (ODM)

The output function of a forms manager is to take the information to be displayed and arrange it in the order dictated by a form. In prior art forms managers, the form of the output is predetermined and stored somewhere in the system. The forms manager does not have the capability to modify the form dependent upon the user or information being displayed.

These types of prior art forms managers have the disadvantage of having to display an information field whether the field is used or not. By way of one specific example, in the area of medical records, there are numerous types of information that can be maintained for any one particular patient However, rarely does one patient require that all of the types of information be maintained. Therefore, it is a disadvantage for the forms manager to display blank spaces when that particular area is not being monitored. By displaying these blank areas, the size of the display is greatly increased making it impossible to display all of the information on one or two screens There may also be screen displays that have no data on them.

In addition, it is often necessary, for record keeping and security, to: limit access to particular portions of the display; keep track of who enters the data; and to insure that the data is entered correctly. Therefore, there is a requirement in the industry that the forms manager also interact with the input side of the systems.

Accordingly, it is an object of the present invention to provide a forms manager which overcomes the above deficiencies.

It is a further object of the present invention to provide a forms manager that acts as an input and output buffer.

Another object of the present invention is to provide a forms manager which customizes the form to the information being displayed.

Still another object of the present invention is to provide a forms manager which utilizes tiles, or windows, to display information.

Yet another object of the present invention is to provide a forms manager which may be used to provide data entry and access security to tiles

Another object of the present invention is to provide a forms manager which operates independently of the applications allowing the forms to be modified without requiring a change in the application.

SUMMARY OF THE INVENTION

A particular embodiment of the present invention consists of a forms manager which first obtains a list of objects to be displayed. The tiles associated with these objects are then obtained from a data base of various tiles. The tiles, and associated objects, are then ranked in order for display. Finally, the tiles and their associated objects and data are displayed.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram representing the function of the present invention within a data processing system;

FIG. 2A is a process flow diagram of a keyboard input process embodying the present invention;

FIG. 2B is a process flow diagram of a pointer input process embodying the present invention;

FIG. 3 is a graphical representation of a display generated by a process embodying the present invention;

FIG. 4 is a graphical representation of an second display generated by a process embodying the present invention;

FIGS. 5A-5C are graphical representations of alternative cell displays in FIGS. 3 and 4;

FIG. 6 is a graphical representation of a display generated by a process embodying the present invention illustrating the pop-up menu function;

FIG. 7 is a process flow diagram of an output buffer process embodying the present invention;

FIG. 8 is a graphical representation of a third display generated by a process embodying the present invention;

FIG. 9 is the display of FIG. 7 after being prioritized by a process embodying the present invention; and

FIG. 10 is a process flow diagram modification to the process flow diagram of FIG. 6.

DETAILED DESCRIPTION OF THE DRAWINGS

As used herein, the term "tile" will refer to a set of rules for displaying data of an object and a list of objects that it may display. A "group" is a collection of related tiles and a "form" is a collection of groups and/or tiles.

A tile is comprised of "cells" each cell having a related "object". The object may be either text, numerical, or a combination thereof. A tile is similar to a window with the exception that tiles do not overlap as is permitted with windows. It should be understood that, if the application permits, the present invention may be utilized with windows and as such the term tile may be taken as synonymous with window. A description of windows is provided in Scheiflen & Gettys, "The X Window System", ACM Transactions on Graphics, Vol. 5, No. 2, pg. 79–109 (April 1986).

Referring initially to FIG. 1, a block diagram, generally designated 10, representing the function of the present invention within a data processing system is illustrated. A forms manager 11 is used as an interface, or buffer, among and between the input, output, and system applications and databases.

An input device 12 is coupled to forms manager 11. Input device 12 is illustrated as a keyboard with a joy stick but may be any type of input device In general, the communication is from input device 12 to forms manager 11. However, there may be a need for forms manager 11 to operate visual or acoustic devices on input device 11. For this reason, there may be some communication from forms manager 11 to input device 12.

Also coupled to forms manager 11 is an operator display module 13. Display 13 is typically a cathode ray tube, but may be any of numerous types of displays such as the various types of flat panel displays Communication is generally from forms manager 11 to display 13. However, a touch sensitive screen or other type of input device may be incorporated in the system requiring communication from display 13 to forms manager 11.

Forms manager 11 also interacts with the application programs 14 and the databases. The databases, for this particular example, are broken into a general database 15 and a specific database 16. It should be understood that while the database may be functionally separated, both may be contained in the same physical location

In operation, objects are entered through input device 12 to the forms manager. This input is checked and stored in database 16 for the particular patient. When a form is to be displayed, forms manager 11 obtains a list of objects to be displayed. One source of objects is application program 14. The forms manager then retrieves the tiles associated with the objects from database 15. The data to be placed in the tiles is then obtained from database 16. Using these inputs, forms manager 11 develops a form which is displayed on display 13.

The input operation of the present invention is illustrated in FIGS. 2A and 2B. A pseudo code listing of the keyboard operation is shown in Appendix A. The line numbers in parentheses refer to code lines in Appendix A. In FIG. 2A, a process flow diagram of a keyboard input process embodying the present invention is illustrated. The process commences with the depression of a key on the keyboard, step 20. The process then proceeds to a decision block 21 where a determination is made as to whether an item on the form has been selected (line 6). If an item has not been selected, the process moves to block 22 where an error message is printed (line 7) and the process exits the routine.

If an item on the form has been selected, the process moves to decision block 23. In decision block 23, the forms manager determines whether the selected item is enterable by the user (line 10). If the item is not enterable by the user, the process moves to block 24 where an error message is printed (line 11). Following the error message, the process exits the routine.

If the item is enterable by the user, the process moves to a decision block 25. In decision block 25, the forms manager determines whether the key was a terminator (line 14) such as a return key, enter key, tab key, or the like. If the key was not a terminator key, the process moves to decision block 26. In decision block 26, the forms manager determines if the key represented a valid character for that item (line 19). By way of example, the item may be limited to numeric input only; in which case any alpha input would be rejected. If the key does not represent a valid character, the process moves to block 27. In block 27 an error message is printed (line 27) and the process exits the routine.

If the key represents a valid character, the process moves to block 29. In block 29, the valid character is appended to an item value for the selected item (line 20–23). The process then exits the routine.

If the key was a terminator, the process moves from decision block 25 to block 28. In block 28 the item value, created in block 29, is stored in a buffer (line 15). Following block 28 is block 30 where a "New Event" signal is generated (line 16) to inform the application that a new item value is present.

A similar process takes place when a pointing device (i.e. mouse, joy stick, roller ball, etc.) is utilized as the input device. This process is illustrated in FIG. 2B which is a flow chart of a pointer input process embodying the present invention. A pseudo code listing for performing this function is contained in Appendix B. The line numbers in parentheses refer to code lines in Appendix B. The processes commences at a block 35 where a pointer input is received.

The process then moves to a decision block 36. In decision block 36, the forms manager determines if the pointer is on an active form (line 5). If the pointer is not in an active form, the process moves to block 37. Block 37 causes an error message to be printed (line 6). The process then exits the routine.

If the pointer is on an active form, the process moves from decision block 36 to decision block 38. In decision block 38 the forms manager determines if the pointer is over a selectable item (line 9). If the item is not selectable, either generally or by that user, the process moves to block 39. Block 39 causes an error message to be printed (line 10). The process then exits the routine.

If the pointer is over a selectable item the process moves from decision block **38** to decision block **40**. In decision block **40**, the forms manager determines whether the pointer is over a pop-up window (line 13). If the pointer is not over a pop-up window, the process moves to a block **41**. In block **41** a "select" event is generated lines (18–27) indicating the item at which the pointer was directed. The select event may permit the entry of data into the item selected or may cause a pop-up menu to be formed.

If the pointer is over a pop-up menu, the process moves from decision block **40** to block **42**. Block **42** causes an event associated with the selected item to be generated (line 14). This event may be the entry of data, the correction of previously entered data, a request to show more detail, or the like. Following block **42**, the process exits the routine.

The process of entering data by the user is classified as an event. More precisely, an event is a user action or set of user actions which are of interest to the application.

The events are placed in a queue by the forms manager, steps **30** and **42** of FIGS. 2A and 2B, respectively. This queue of events is maintained by the forms manager until acted upon by the applications program. Periodically, the application program will request the next event from the forms manager.

In the output operation, the forms manager is directed to display a particular form containing objects (text/data). In the present invention, the text/data to be displayed dictate the shape of the form rather than the form dictating how the objects are displayed. A sample form, generally designated **50**, is illustrated in FIG. 3. Form **50** illustrates two groups of tiles, a vitals group **51** and a labs group **52**. Vitals group **51** contains four tiles: a heart rate tile **53**, a blood pressure tile **54**, a temperature tile **55**, and a respiration rate tile **56**. The labs group consists of: a CBC/HGB (Complete Blood Count-/HemoGloBin) tile **57**, a CBC/HCT (Complete Blood Count/HematoCriT) tile **58**, a K+(Potassium) tile **59**, a Na (Sodium) tile **60**, a BUN (Blood Urea Nitrogen) tile **61**, and a CREAT (CREATinine) tile **62**.

Each tile **53–62** comprises several cells. For example, heart rate tile **53** has a group cell **53A** and a title cell **53B**. In a group, such as group **51**, the name of the group may or may not be printed. Generally, only the first tile in the group, here tile **53**, will designate the group name. Cell **53B** contains the title; here the title is "Heart Rate". The remaining cells of tile **53** are data cells **53C**. One cell **53C** is provided for every time period listed on the display. Cells **53C** are spreadsheet type cells in that they repeat with time whereas regions **53A** and **53B** are non-spreadsheet cells that do not repeat with time. In tile **54**, the group cell **54A** is left blank since the group is displayed previously. However, the object cell **54B** of tile **54** is labeled Blood Pressure. The data cells **54C** of tile **54** are then aligned with the time intervals.

There is generally more than one type of tile for each category. One tile **53** may be provided to display its regions in either a horizontal fashion, as in FIG. 3, or in a vertical fashion by a tile **53'**, as shown in FIG. 4. In this particular embodiment, the tiles are formed of rectangles. Each region in a tile will have the same height in horizontal tiles and the same width in vertical tiles. The height (width) of a horizontal (vertical) tile is determined by the space required by the largest region of

the tile. Typically, this cell will be the object region, such as region **54'B**, FIG. 4.

In addition to the two standard types of tiles (horizontal and vertical) there may be other tiles displaying the data in different fashions. For example, blood pressure cell **54C** will typically display three types of data, X, Y, and Z, representing the high pressure (X), low pressure (Y), and the average pressure (Z). In FIGS. 5A–5C, three different ways are illustrated to display this data. The dimensions of the cells and tiles will vary depending on the display desired

Each tile may be designed to contain numerous types of information, both displayed and not displayed. For example, the basic type of information is whether a cell contains data or text. A text (data) cell of a tile can be set to prohibit the entrance of data (text). The physical size and shape of the cell and tile is also maintained in the tile definition. The display rules are in the tile definition indicating, for example, which cell display format (refer to FIGS. 5A–5C) will be used. Each tile also has an associated name and/or ID number with which it is referenced.

Various security measures are also contained within the tile/cells descriptions. These security measures take the form of whether a tile/cell may be accessed and by whom. For example, you would not ordinarily allow access to the group and title cells. These are preset and are to be changed only by the system administrator. In other areas, such as the data areas, you want the general nursing population of the hospital to be able to enter data. However, it would also be desirous to know who entered the data and exactly when it was entered. While the charts may be broken down into half-hour segments, the vitals may not be taken right on the half-hour, so you will want to keep a record of the exact time and who entered the data. An additional security measure is whether data may be changed once it has been entered. While you do not want to have data changed at a later date, it is important that the data illustrated is accurate. Therefore, a tile/cell will retain the original entry and who made it; and the changed data and who made the changes. When a cell is displayed that has had data changed, this region will be offset from the other data by a marker, different color, flashing display, or the like.

The time of the data may be broken down further into when the readings were made and when they were actually entered into the system. The tile/cell definitions may also perform a data integrity check to ensure the entered data is not obviously incorrect. For example, in temperature tile **55**, if the temperature 986° C. is accidently entered rather than 98.6° C., the system will recognize this as an error and alert the individual making the entry. In addition, if an entry is outside a normal range, but still valid, this entry may be offset from the rest of the data in a fashion described above with respect to the changed data.

In this particular embodiment, tiles are designed to display the various forms of objects. There is not one tile for each object but one tile for each form of object. When the forms manager seeks to display a particular object, it determines which tile is associated with this object and retrieves that tile. As an example, Potassium (K+) tile **59** and Sodium (Na) tile **60** would be the same form of tile. The difference would be the title displayed in title cells **59B** and **60B** and the data entered into spreadsheet data cells **59C** and **60C**.

Another feature of the tiles of the present invention is that a cell of a tile may be associated with a menu of possible actions with event codes. An example of this is provided in FIG. 6. In FIG. 6, the operator has selected cell 65 of temperature tile 55. When this is selected, a pop-up menu will appear, such as menu 70, having the possible actions which may be undertaken in cell 65 of tile 55.

As shown in FIG. 6, menu 70 is illustrated with three options. The first event is to make an entry. This is selected when an entry is first being made to the cell. The second event is to correct an entry. This is selected if the cell has been previously accessed. The third region is labeled "SHOW DETAIL". This region is selected to show the details such as the time of entry, the person who made the entry, or other similar information. As an alternative to the pop-up menus, a series of soft keys may be utilized. This would provide a pop-up, or existing menu window, with a list of keys, generally function keys, that are assigned for one of the events listed above. These soft keys may have different functions assigned by the particular region being accessed A process for generating a form embodying the present invention is illustrated in the flow chart of FIG. 7. This process is illustrated in pseudo code in Appendix C. The following lines contained in parentheses refer to the line numbers of Appendix C. Initially, a request, box 80, for a new form is received by the process (line 3). The system then goes to decision box 81 and decides if there is an object list in existence for this particular form (line If there is no list of objects associated with this form, the process moves to step 82 where an object list is built from the form definition (line 10). An object list may come from three different sources: a list associated with the form; a list entered at the time of the form generation request; or a list provided by an application program.

Once the object list has been generated, the process moves on to step 83 where objects are assigned to tiles (lines 12, 13). For example, if the object is the "Heart Rate", then the tiles would be checked to see which tile was associated with "Heart Rate".

Next, in step 84, the draw point is moved to the top, or some designated home point, of the display (lines 15–17). A decision is then made in step 85 as to whether there are objects/tiles to be drawn (lines 19–20). If there are no tiles to be drawn, the process moves to step 86 and ends. If there are tiles to be drawn, the process moves to step 87 where the object/tile is drawn (lines 21–54). Once drawn, the process moves to step 88 where the draw point is moved to the next position (line 55).

This process would generate a display such as that illustrated in FIG. 3. However, there are generally so many objects which may be monitored for a particular patient, that not all of the objects may be displayed in one screen. Since it is unnecessary to display objects which are not being monitored, the elimination of these objects will leave more room on the display for objects that are being monitored.

As an example, of the objects displayed in FIG. 3, if

temperature, CBC/HGB, and CBC/HCT are not being monitored, then display space is being taken up with empty, unutilized space. To avoid this and to provide a more efficient display, the forms manager will not display items that are not being monitored. The display would then appear as shown in FIG. 8. Here, the three objects that are not being utilized have been eliminated and the form adjusted to remove the tiles associated with those objects. This will leave room at the bottom of the display for additional tiles that were not displayed in the form of FIG. 3.

In addition to the above, the forms manager will place the objects in any order specified in the listing of objects that is provided or generated. For example, if a particular doctor prefers to view the blood pressure, tile 54, as the first object, this can be indicated in the form description, and the forms manager, when the display is created, will place this tile first. In addition, the forms manager may be set to inter-mix objects from the various groups if desired. An example of this is illustrated in FIG. 9.

In FIG. 9, the tiles have been arranged by the forms manager in order of a priority that was provided with the description of the form. In a medical setting, there may be one generally form for the hospital used by the nursing staff and a different form for doctors. Further, particularly for specialists who do not desire to see everything or who desire to see certain information (objects) first, there may be forms established for each doctor.

Therefore, the process of FIG. 7 may be provided with the additional steps shown in FIG. 10. These steps may be inserted between decision step 81 and step 84 and may be in any order desired. Following step 81 is step 89 in which any objects which are not being monitored from the object list are eliminated from the object list. Next, in decision step 90, the process looks to see if there is a priority list associated with the form. If there is no priority list, the process moves to step 91 where a default priority list is used to arrange the objects. If there is a priority list for the form, the process moves to step 92 where the objects are arranged according to the priority list. The process then moves to step 83.

As indicated above, the assignment of objects to tiles, in step 83, may occur before, during, or after the steps set out in FIG. 10.

Therefore, as stated above, the present invention creates a form to fit the objects being displayed rather than fitting the objects to the form.

Thus, it will be apparent to one skilled in the art that there has been provided in accordance with the invention, a process and method that fully satisfy the objects, aims, and advantages set forth above.

While the invention has been described in conjunction with specific embodiments thereof, it is evident that many alterations, modifications, and variations will be apparent to those skilled in the art in light of the foregoing description. Accordingly, it is intended to embrace all such alterations, modifications, and variations in the appended claims.

APPENDIX A

```
fm_click(key)

  if item not selected
     error - "Keyboard inactive when nothing selected"
     exit

  if selected item not enterable
     error - "Item not enterable"
     exit

  if key is a terminator
     store item value in database
     generate NEW event
     exit

  else if key is a valid key
     if this is the first key in this item
        set item value to blank

     add key to item value
     exit

  else
     error - "Invalid key"
     exit
```

APPENDIX B

```
fm_point (position)

  if position not on current form
     error - "Not on active form"
     exit

  if position not on selectable item
     error - "Item not selectable"
     exit

  if position over popup menu
     generate action associated with popup menu item
     exit

  else
     if there is a currently selected item
        if position is over selected item
           generate DESELECT action for item

        else
           generate DESELECT action for selected item
        generate SELECT action for item at position

    else
       generate SELECT action for item at position

    exit
```

APPENDIX C

```
fm_new (form_name, object_list)

  read form definition

  if object list supplied
     make local copy of object list
  else
     create object list from form definition

  for each object in object list
     find tile in form which displays object

  draw point = (0,0)
  last tile = NONE
  last group = NONE

  for each object on object list
  do
     set this tile to tile containg object
     set this group to group containing tile

     if this group != last group

        if last tile != NONE
           draw end of last tile
           draw point = end of last tile

        if last group != NONE
           draw end of last group
           draw point = end of last group

        last group = this group
        last tile = this tile

        draw beginning of this group
        draw point = beginning of this group

        draw beginning of this tile
        draw point = beginning of this tile

     else if this tile != last tile

        if last tile != NONE
           draw end of last tile
           draw point = end of last tile

```